

García Argueta Jaime Daniel, 312104739, jaimegarciaargueta@ciencias.unam.mx
Flores González Luis Brandon, 312218342, iluis@ciencias.unam.mx

Justificación del ejercicio 1

- a) Traduce el siguiente enunciado: Todos los estudiantes molestan a los que les enseñan, cuando estos los reprueban.

$E(x)$ = x es estudiante $P(x)$ = x es profesor $M(x, y)$ = x molesta a y
 $R(x, y)$ = x reprueba a y $\tilde{N}(x, y)$ = x enseña a y

Para todo x, Para todo y ($E(x) \wedge P(y) \wedge R(y, x) \wedge \tilde{N}(y, x) \rightarrow M(x, y)$)

- b) Construye una expresión que represente el enunciado anterior:

$(\text{All "x"} \text{ All "y"} (\text{Imp} (\text{Conj} (\text{Conj} (\text{Pr "E"} [V "x"]) (\text{Pr "P"} [V "y"]))) (\text{Conj} (\text{Pr "R"} [V "y", V "x"]) (\text{Pr "\tilde{N}}" [V "y", V "x"])))) (\text{Pr "M"} [V "x", V "y"])))$

- c) Construye un mundo en el cual el enunciado sea cierto.

$U = \{\text{Favio, Susana, Fernando}\}$ $I(E) = \{\}$

Como el antecedente no se cumple, ya que $I(E)$ es vacía. Entonces concluimos que el enunciado es cierto.

- d) Construye un mundo en el cual el enunciado sea falso.

$U = \{\text{Luis, Alexis, Javier, Favio, Susana, Fernando}\}$
 $I(E) = \{\text{Luis, Alexis, Javier}\}$
 $I(P) = \{\text{Favio, Susana, Fernando}\}$
 $I(M) = \{\}$
 $I(R) = \{(\text{Favio, Luis}), (\text{Favio, Alexis}), (\text{Favio, Javier}), \dots, (\text{Fernando, Luis}), (\text{Fernando, Alexis}), (\text{Fernando, Javier})\}$
 $I(\tilde{N}) = \{(\text{Favio, Luis}), (\text{Favio, Alexis}), (\text{Favio, Javier}), \dots, (\text{Fernando, Luis}), (\text{Fernando, Alexis}), (\text{Fernando, Javier})\}$

El antecedente se cumple ya que todos los profesores reprueban a los estudiantes, todos los profesores enseñan a los estudiantes, son todos los estudiantes y son todos los profesores. Por otro lado el consecuente no se cumple ya que es un conjunto vacío. Por lo tanto el enunciado es falso.

Después se importan los módulos LPO, LPOPSemantica y Semanal2 en el mismo directorio que Practica3. Que previamente se desarrollaron.

Se crea la **función simpSus**, la cual simplifica una lista de sustituciones eliminando las del tipo $(x, \vee x)$ ($x:=x$).

Se crea la **función compSus**, la cual devuelve la composición de las dos funciones dadas (primero se aplica la del primer parámetro y después la del segundo parámetro).

Se crea la **función unifica**, la cual obtiene el unificador más general de dos términos si es que este existe y si no devuelve la lista vacía. Se usa la **función auxiliar verifOcurr**, **apSusTList** y **comp**.

Se crea la función **unificaConj**, la cual obtiene el unificador más general de un conjunto de términos si es que este existe, si no entonces devuelve la lista vacía.

Se crea la **función unificaListas**, la cual Unifica las listas de términos de una función término a término y luego las compone.

Se crea la **función unificaLit**, la cual obtiene el unificador más general de dos literales, en caso de que no sea posible unificar es regresada la lista vacía. Se usa la **función auxiliar apSusTList** y **comp**.

Se necesitaron agregar tres funciones auxiliares, las cuales están bien documentadas en el código. Quiero aclarar que se hizo todo lo que se pide en la práctica.